UNITED STATES PATENT APPLICATION

FOR

DAISY CHAIN LATENCY REDUCTION

Inventors:

Davis S. Bormann
Kirk W. Skeba

DAISY CHAIN LATENCY REDUCTION

BACKGROUND OF THE INVENTION

1. FIELD OF INVENTION

[0001] This invention relates generally to high-speed serial bus operation, and more specifically to low latency techniques applied to a multi-link serial bus architecture.

2. ART BACKGROUND

[0002] A data bus is commonly used in data processing devices such as a computer to send and receive data from a plurality of devices. Data buses have been constructed around a serial or a parallel architecture. For example, a serial data bus can connect together a printer, a scanner, a storage device, and a wireless transceiver to facilitate communication between devices. Such a configuration of devices connected sequentially is known in the art as a daisy chain.

[0003] A controller is used to send data to a first device on the serial data bus. Data prepared for transmission on a serial data bus can be referred to in the art as a packet of data. The packet of data typically has an address corresponding to a unique device on the serial data bus. The data stream transmitted onto the serial data bus can consist of a sequence of packets randomly addressed to different devices distributed along the serial data bus.

[0004] In serial data bus architecture, a plurality of bits (a bit stream) is transmitted sequentially in time across a data transmission path. Thus, a first bit

is followed by a second bit, etc., eventually completing the transmission of a first data word and then a second data word. Data transfer accomplished using serial data bus architecture has traditionally suffered from high latency. Current serial data bus protocol requires the transmission of the serial data word or packet to the first device in the daisy chain and then from the first device to the second device in the daisy chain, progressing down the serial data bus to the last device. According to present bus architecture, each device along the serial data bus reads the packet of data, checks it for errors and then retransmits the packet to the next device if the packet address did not match the given device address. If an error is found, the device notifies the controller along a second serial data transmission path and the controller responds to the error condition.

[0005] Such a protocol suffers from high latency, since each device must read the data and perform an error check on the packet of data, even if the packet of data is not addressed to the device performing the check.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present invention is illustrated by way of example and is not limited in the figures of the accompanying drawings, in which like references indicate similar elements.

[0007] **Figure 1** illustrates a serial data word or packet of data.

[0008] **Figure 2** illustrates a serial data bus architecture utilizing dual serial data transmission paths.

[0009] **Figure 3** shows a flow chart for a method according to one embodiment.

DETAILED DESCRIPTION

[0010] In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and specific embodiments in which the invention may be practiced are shown by way of illustration. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the invention is defined only by the appended claims.

[0011] A low latency high-speed serial data bus is disclosed that reduces latency by 90% (in one embodiment) over present serial data bus protocols. Error checking is postponed until the serial data word or data packet reaches the destination device. Along the serial data transmission path a device only reads as much of the address of the serial data word as is necessary for the device to determine if the serial data word is addressed to the device. The device does not read the entire data word if the device is not the addressee of the serial data word. Latency is reduced accordingly, since devices in between the controller and the destination device do not read the entire data word. Once the serial data word has arrived at the destination device, the destination device reads the entire serial data word address and determines a match; then the device reads the data word. Error checking occurs and if an error condition is found, the device notifies the controller via a second data transmission path. In one embodiment, a 20-bit serial data word has a 2-bit address. Reading the 2-bit address instead of

reading the entire 20 bit serial data word and performing error checking reduces the latency of the data transfer by at least 90%.

[0012] A serial data word or packet is shown in **Figure 1** at 100. The serial data word 100 includes a device address 102 and device data 104. The device address can be one or more bits in length and is displayed generally in **Figure 1** as having 1 to i bits. The number of bits in the address field determines the number of devices that can be daisy chained together to form the serial data path. For example, a 2-bit address field would support four devices and a 4-bit address field would support sixteen devices. The length of the address field can be adjusted according to the requirements of a particular design; the present invention is not limited by the choice of the length of the address field. Device data 104 is illustrated in **Figure 1** as containing 1 to j bits. Serial data word 100 is therefore k total bits in length as indicated by 106, where k represents a sum of i and j bits. The size of the device data field 104 is variable and can be adjusted according to the requirements of a given system design; the present invention is not limited by the number of bits selected for the length of the device data field. Neither does the number of total bits in the serial data word 100 limit the present invention.

[0013] Additional fields can be included in serial data word 100. For example, in one embodiment it may be desirable to include an endpoint number and a parity bit as well as a type indicator. In one embodiment, an 8-bit end point number, and an 8-bit data field are used. The serial data word or packet may be assembled in any order. In order to achieve the latency reduction, according to

the teaching of the present invention, the only requirement on the serial data word is that the device read the address of the serial data word before reading the device data 104. Reading the device address 102 before the device data 104 allows the device to retransmit the serial data word to the next link in the serial data path (daisy chain) without expending time reading the device data when the serial data word's destination is not the device. In one embodiment, the serial data word has a data structure that uses at least two address bits, followed by a type indicator bit, followed by an 8-bit end point number, followed by an 8-bit data field, and ending with a parity bit.

[0014] **Figure 2** illustrates a serial data bus architecture utilizing dual serial data transmission paths. With reference to **Figure 2**, a 1st serial data transmission path includes a 1st link 202a, a second link 202b, up to and including an *n* th link 202c. The serial data transmission path 202 can include a general number of links *n*, limited only by the design of the particular data processing system. The present invention does not impose a limit on the number of devices *n*.

[0015] In one embodiment, a controller 206 transmits a serial data word onto the first link 202a, the serial data word is addressed to one of the devices daisy chained along the 1st serial data transmission path 202. For example, 1st device 208, 2nd device 210, or *n* th device 212. The 1st device 208 connects the 1st link 202a with the second link 202b. The 2nd device 210 connects 2nd link 202b with *n* th link 202c. The *n* th link 202c terminates into the last device along the daisy chain, *n* th device 212. If the serial data word transmitted from the

controller 206 is not addressed to the 1st device 208, only enough of the device address attached to the serial data word is read by the 1st device 208. After determining that the serial data word is not addressed to the 1st device 208 the 1st device 208 retransmits the serial data word to the 2nd device 210 across the 2nd link 202b. If the serial data word is addressed to the 2nd device 210, then the 2nd device 210 reads the rest of the serial data word. If the serial data word was not addressed to the 2nd device 210 then the serial data word would be retransmitted by successive devices until the serial data word reached the device to which it was addressed.

[0016] Assuming for this discussion that the 2nd device is the device to which the serial data word was addressed, the 2nd device would then perform error checking and notify the controller 206 if an error condition was in existence. Notification of the controller 206 is accomplished using a 2nd serial data transmission path 204. The second serial data transmission path 204 includes a first link 204a and a 2nd link 204b up to and including an n^{th} link 204c. The second serial data transmission path 204 is similar to the first data serial data transmission path 202, except that the communication occurs in reverse order across the second serial data transmission path with respect to the first serial data transmission path. Together, both serial data communication paths 202 and 204 form a bi-directional communications link that enables communication to and from the controller and all devices that are daisy chained together.

[0017] **Figure 3** shows a flow chart for a method according to one embodiment of the present invention, which is consistent with the description of

Figure 2 discussed previously. With reference to **Figure 3**, transmitting a first serial data word on a first link of a 1st serial data transmission path is indicated at 302. The process block 302 describes the communication between the controller 206 and the 1st link 202a in **Figure 2**. In process block 304 of **Figure 3**, a 1st device reads a device address of the serial data word. As mentioned previously, the device 208 (**Figure 2**) need only read enough of the address as is required for the device 208 to determine whether or not the serial data word is addressed to the device 208. In another embodiment, the device can be configured to read the entire device address 102 (**Figure 1**) of the serial data word.

[0018] A next process block 306 (**Figure 3**) describes passing the serial data word to another device along the serial data communication path. If the device address 102 (**Figure 1**) does not match an address of the 1st device 208 (**Figure 2**) then the serial data word is passed to the next device according to process block 306 (**Figure 3**). For example, the serial data word could be passed to the second device 210 from the 1st device 208 (**Figure 2**).

[0019] Process block 308 (**Figure 3**) indicates the first time that the entire serial data word will be read by a device. The device data 104 (**Figure 1**) in the serial data word 100 is read by a device when the device address 102 matches an address of a device connected to the serial data transmission path 200 (**Figure 2**). After the serial data word reaches the correct device, the device checks for an error condition at 310. If an error condition is found, then the device notifies the controller 206 (**Figure 2**) via the 2nd serial data communications path 204. The controller can then perform the required analysis

of the error condition and respond by resending the serial data word or taking other appropriate action.

[0020] One or more of the links of the serial data communication path 202a, 202b, 202c, 204a, 204b, and 204c can be wireless links, or the links can be made of a metallic conductor, such as copper. The present invention is not limited by the energy transmission medium used for the links of the serial data transmission path. In one embodiment, all of the links can be made of a metallic conductor. In another embodiment a pair of links (for example, 202a and 204a) could be wireless links with the remaining links being made out of metallic conductors. Alternatively, a pair of links (for example, 202a and 204a) could be made out of metallic conductors and the balance of the links could be wireless. The present invention can be used with a point of entry into a wireless communications network. Any combination is possible; the present invention is not limited by the combination of metallic and non-metallic conductors used for the links of the serial data transmission path.

[0021] The present invention can be used in a variety of data transmission applications. For example the present invention is generally applicable in high-speed serial data bus scenarios.

[0022] In this description of the invention the term "device" has been used to describe the devices that are daisy chained together along the serial data transmission path. As used herein, the term "device" is generic, providing the functionality of receiving the serial data word, reading the serial data word and retransmitting the serial data word. The device can be referred to as a

communications front end. Alternatively, the device can represent components of a data processing system such as a communications front end, a data storage unit, a wireless device, a local area network, a cellular telephone, and an entry point. The devices listed above can incorporate the necessary transmit and receive electronics necessary to communicate with the serial data transmission path, as is well known in the art. For example, a storage device can be configured with network capability that allows it to be daisy chained together with links 202a and 204a, as described in conjunction with **Figure 2.**

[0023] It will be appreciated that the methods described in conjunction with the figures may be embodied in machine-executable instructions, e.g. software. The instructions can be used to cause a general-purpose or special-purpose processor that is programmed with the instructions to perform the operations described. Alternatively, the operations might be performed by specific hardware components that contain hardwired logic for performing the operations, or by any combination of programmed computer components and custom hardware components. The methods may be provided as a computer program product that may include a machine-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform the methods. For the purposes of this specification, the terms "machine-readable medium" shall be taken to include any medium that is capable of storing or encoding a sequence of instructions for execution by the machine and that cause the machine to perform any one of the methodologies of the present invention. The term "machine-readable medium" shall accordingly be taken to

included, but not be limited to, solid-state memories, optical and magnetic disks, and carrier wave signals. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or a produce a result.

[0024] Thus, a novel method for reducing latency in a high-speed serial data bus is described. Although the invention is described herein with reference to specific preferred embodiments, many modifications therein will readily occur to those skilled in the art. Accordingly, all such variations and modifications are included within the intended scope of the invention as defined by the following claims.